REMARKS

Claim 2-21 are pending in this application.

Claims 2 and 12 have been rejected.

Claims 20-21 have been allowed.

Claims 3-11 and 13-19 are identified as being allowable if rewritten in independent form.

Claims 12-20 have been amended.

Claims 2-21 remain in the application.

The Applicant has attached an appendix containing all pending claims for the Examiner's convenience.

I. ALLOWABLE CLAIMS

The Applicant thanks the Examiner for the indication that Claims 20-21 are allowable. Claim 20 has been amended but remains in condition for allowance. The Applicant also thanks the Examiner for the indication that Claims 3-11 and 13-19 would be allowable if rewritten in independent form. Because the Applicant believes that Claims 3-11 and 13-19 depend from allowable base claims, the Applicant has not rewritten Claims 3-11 and 13-19 in independent form.

II. 35 U.S.C. § 103 – Obviousness

The Office Action rejects Claims 2 and 12 under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,475,823 by Amerson et al. ("*Amerson*") in view of U.S. Patent No. 6,282,633 by Killian et al. ("*Killian*"). This rejection is respectfully traversed.

In *ex parte* examination of patent applications, the Patent Office bears the burden of establishing a *prima facie* case of obviousness. MPEP § 2142; *In re Fritch*, 972 F.2d 1260, 1262, 23 U.S.P.Q.2d 1780, 1783 (Fed. Cir. 1992). The initial burden of establishing a *prima facie* basis to deny patentability to a claimed invention is always upon the Patent Office. MPEP § 2142; *In re Oetiker*, 977 F.2d 1443, 1445, 24 U.S.P.Q.2d 1443, 1444 (Fed. Cir. 1992); *In re Piasecki*, 745 F.2d 1468, 1472, 223 U.S.P.Q. 785, 788 (Fed. Cir. 1984). Only when a *prima facie* case of obviousness is established does the burden shift to the applicant to produce evidence of nonobviousness. MPEP § 2142; *In re Oetiker*, 977 F.2d 1443, 1445, 24 U.S.P.Q.2d 1443, 1444 (Fed. Cir. 1992); *In re Rijckaert*, 9 F.3d 1531, 1532, 28 U.S.P.Q.2d 1955, 1956 (Fed. Cir. 1993). If the Patent Office does not produce a *prima facie* case of unpatentability, then without more the applicant is entitled to grant of a patent. *In re Oetiker*, 977 F.2d 1443, 1445, 24 U.S.P.Q.2d 1443, 1444 (Fed. Cir. 1992); *In re Grabiak*, 769 F.2d 729, 733, 226 U.S.P.Q. 870, 873 (Fed. Cir. 1985).

A *prima facie* case of obviousness is established when the teachings of the prior art itself suggest the claimed subject matter to a person of ordinary skill in the art. *In re Bell*, 991 F.2d

781, 783, 26 U.S.P.Q.2d 1529, 1531 (Fed. Cir. 1993). To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed invention and the reasonable expectation of success must both be found in the prior art, and not based on applicant's disclosure. MPEP § 2142.

The Office Action acknowledges that *Amerson* fails to disclose detecting an instruction "without computing an external memory address of [a] first memory location" as recited in Claims 2 and 12. The Office Action relies on *Killian* as disclosing this element.

Killian recites a RISC processor. (*Abstract*). The processor includes an "execution and address calculation" or "E" stage. (*Col. 4, Lines 41-44*). This stages computes a "virtual address" of a memory location. (*Col. 6, Lines 43-47*). The virtual address is the same as the external address for the memory location. (*Col. 7, Lines 33-37*). The processor uses a portion of the virtual address to perform a "special bypass." (*Col. 7, Lines 10-23*).

Killian lacks any mention of determining whether a "special bypass" should be performed without computing an external memory address. In fact, *Killian* expressly teaches that an external memory address is computed and used to determine whether a "special bypass" should be performed.

As a result, *Killian* fails to disclose, teach, or suggest detecting an instruction "without computing an external memory address of [a] first memory location" as recited in Claims 2 and 12.

The Office Action appears to impermissibly characterize this limitation of Claims 2 and 12. In particular, the Office Action states that "based on applicant's own disclosure the feature of without computing the address was directed to the comparison of the identical offset or registers .... Therefore, it is assumed that a portion, or segment of the address could be used for comparison purpose." (*Office Action, Page 2, Last paragraph*). The Office Action then appears to reject Claims 2 and 12 on the basis that *Killian* discloses using a "portion" or "segment" of an external memory address to identify a particular instruction.

The Applicant respectfully notes that Claims 2 and 12 clearly and unambiguously recite detecting an instruction "without computing an external memory address of [a] first memory location." As a result, any rejection must be based on the cited references disclosing, teaching, or suggesting the detection of an instruction "without computing an external memory address of [a] first memory location" as recited in Claims 2 and 12. Because the Office Action rejects Claims 2 and 12 on a basis other than the actual claim recitations, the Applicant respectfully notes that this is an improper § 103(a) rejection.

For the reasons set forth above, the Applicant respectfully requests withdrawal of the § 103(a) rejection and full allowance of Claims 2 and 12.

SUMMARY

For the reasons given above, the Applicant respectfully requests reconsideration and allowance of the pending claims and that this patent application be passed to issue. If any outstanding issues remain, or if the Examiner has any suggestions for expediting allowance of this patent application, the Applicant respectfully invites the Examiner to contact the undersigned at the telephone number indicated below or at ***wmunck@davismunck.com***. The Applicant respectfully denies any position or averment of the Examiner that is not specifically addressed by the foregoing argument and response.

Respectfully submitted,

DAVIS MUNCK, P.C.

Date: Feb. 11, 2003

William A. Munck
Registration No. 39,308

Docket Clerk
P.O. Drawer 800889
Dallas, Texas 75380
Tel: (972) 628-3600
Fax: (972) 628-3616
Email: *wmunck@davismunck.com*

APPENDIX

2. (Unchanged) A pipelined microprocessor capable of detecting an instruction that loads data from a first memory location that was previously stored to without computing an external memory address of said first memory location.

3. (Unchanged) A pipelined microprocessor as claimed in Claim 2 wherein said pipelined microprocessor is capable of detecting an instruction that stores data into a second memory location that was previously read from without computing an external memory address of said second memory location.

4. (Unchanged) A pipelined microprocessor as claimed in Claim 2 wherein said pipelined microprocessor is capable of detecting instructions that load data from identical memory locations that were previously stored to without computing external memory addresses of said identical memory locations.

5. (Unchanged) A pipelined microprocessor as claimed in Claim 3 wherein said pipelined microprocessor is capable of detecting instructions that store data into identical memory locations that were previously read from without computing external memory addresses of said identical memory locations.

6. (Unchanged) A pipelined microprocessor as claimed in Claim 4 wherein said pipelined microprocessor is capable of examining symbolic structure of said instructions that load data from identical memory locations that were previously stored to, and capable of detecting said instructions that load data from identical memory locations by examining said symbolic structure.

7. (Unchanged) A pipelined microprocessor as claimed in Claim 5 wherein said pipelined microprocessor is capable of examining symbolic structure of said instructions that store data into identical memory locations that were previously read from, and capable of detecting said instructions that store data into identical memory locations by examining said symbolic structure.

8. (Unchanged) A pipelined microprocessor as claimed in Claim 6 wherein said pipelined microprocessor is capable of detecting said instructions that load data from identical memory locations by identifying an identical offset address value from an identical base address value in a register within said pipelined microprocessor.

9. (Unchanged) A pipelined microprocessor as claimed in Claim 7 wherein said pipelined microprocessor is capable of detecting said instructions that store data into identical memory locations by identifying an identical offset address value from an identical base address value in a register within said pipelined microprocessor.

10. (Unchanged) A pipelined microprocessor as claimed in Claim 6 wherein said pipelined microprocessor comprises:

an instruction decode stage capable of detecting said instructions that load data from identical memory locations by identifying an identical offset address value from an identical base address value in a register within said pipelined microprocessor; and

a bypass element capable of sending a bypass signal to an instruction execution stage of said pipelined microprocessor that indicates that said instructions refer to an identical memory location.

11. (Unchanged) A pipelined microprocessor as claimed in Claim 7 wherein said pipelined microprocessor comprises:

an instruction decode stage capable of detecting said instructions that store data into identical memory locations by identifying an identical offset address value from an identical base address value in a register within said pipelined microprocessor; and

a bypass element capable of sending a bypass signal to an instruction execution stage of said pipelined microprocessor that indicates that said instructions refer to an identical memory location.

12. (Amended) A method for operating a pipelined microprocessor, said method comprising **[the step of]**:

detecting in said pipelined microprocessor an instruction that loads data from a first memory location that was previously stored to without computing an external memory address of said first memory location.

13. (Amended) A method for operating a pipelined microprocessor as claimed in Claim 12, said method further comprising **[the step of]**:

detecting in said pipelined microprocessor an instruction that stores data into a second memory location that was previously read from without computing an external memory address of said second memory location.

14. (Amended) A method for operating a pipelined microprocessor as claimed in Claim 12, said method further comprising **[the step of]**:

detecting in said pipelined microprocessor instructions that load data from identical memory locations that were previously stored to without computing external memory addresses of said identical memory locations.

15. (Amended) A method for operating a pipelined microprocessor as claimed in Claim 13, said method further comprising **[the step of]**:

detecting in said pipelined microprocessor instructions that store data into identical memory locations that were previously read from without computing external memory addresses of said identical memory locations.

16. (Amended) A method for operating a pipelined microprocessor as claimed in Claim 14, said method further comprising **[the steps of]**:

examining in said pipelined microprocessor symbolic structure of said instructions that load data from identical memory locations that were previously stored to; and

detecting said instructions that load data from identical memory locations by examining said symbolic structure.

17. (Amended) A method for operating a pipelined microprocessor as claimed in Claim 15, said method further comprising **[the steps of]**:

examining in said pipelined microprocessor symbolic structure of said instructions that store data into identical memory locations that were previously read from; and

detecting said instructions that store data into identical memory locations by examining said symbolic structure.

18. (Amended) A method for operating a pipelined microprocessor as claimed in Claim 16, said method further comprising **[the steps of]**:

detecting in an instruction decode stage of said pipelined microprocessor said instructions that load data from identical memory locations by identifying an identical offset address value from an identical base address value in a register within said pipelined microprocessor; and

sending a bypass signal from a bypass element to an instruction execution stage of said pipelined microprocessor wherein said bypass signal indicates that said instructions refer to an identical memory location.

19. (Amended) A method for operating a pipelined microprocessor as claimed in Claim 17, said method further comprising **[the steps of]**:

detecting in an instruction decode stage of said pipelined microprocessor said instructions that store data into identical memory locations by identifying an identical offset address value from an identical base address value in a register within said pipelined microprocessor; and

sending a bypass signal from a bypass element to an instruction execution stage of said pipelined microprocessor wherein said bypass signal indicates that said instructions refer to an identical memory location.

20. (Amended) A method for operating a pipelined microprocessor, said method comprising **[the steps of]**:

detecting a first instruction that stores data to a first memory location, said first instruction comprising syntax for computing an effective address for said first memory location;

detecting a second instruction that loads data from a second memory location, said second instruction comprising syntax for computing an effective address for said second memory location;

determining said syntax for said first instruction and said syntax for said second instruction;

using said syntax for said first instruction and said syntax for said second instruction to determine a relationship between said first memory location and said second memory location, without computing said effective address for said first memory location and without computing said effective address for said second memory location; and

using said relationship to determine whether to perform one of said first instruction and said second instruction.

21. (Unchanged) A method for operating a pipelined microprocessor as claimed in Claim 20 wherein said syntax for said first instruction and said syntax for said second instruction refer to an identical memory location.